

Forschungspraktikum am ITIV

Kontakt am ITIV: forschungspraktikum@itiv.kit.edu
<http://www.etit.kit.edu/forschungspraktikum.php>

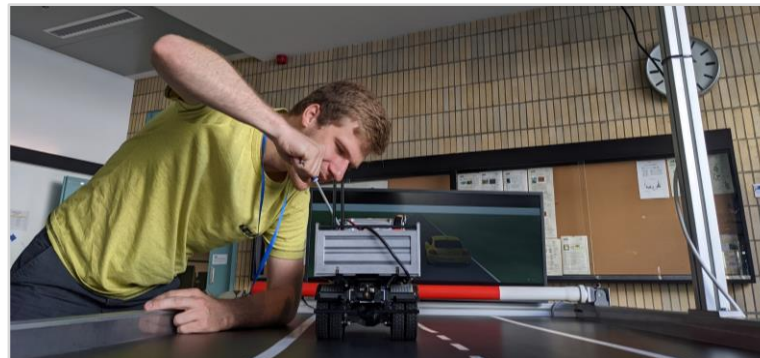
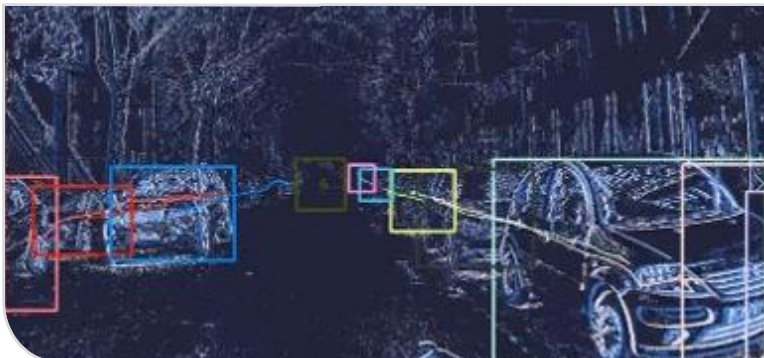


Forschungspraktikum am ITIV

Am ITIV bieten wir eine Vielzahl von interessanten und interdisziplinären Themen für das Forschungspraktikum an.

In den folgenden Folien sind einige dieser Themen dargestellt.

Bei Interesse wenden Sie sich bitte an
forschungspraktikum@itiv.kit.edu.



Forschungspraktikum am ITIV

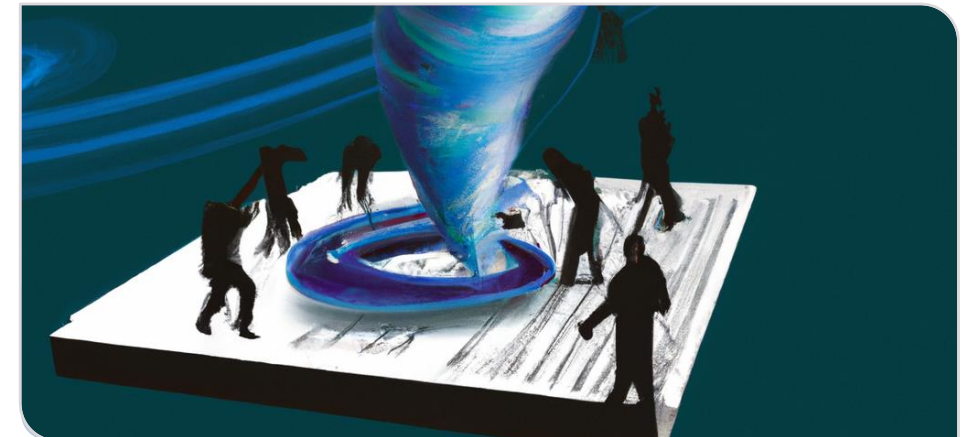
- **Arbeitsumfang**
 - 12 Wochen
 - Mindestens 35 Stunden pro Woche
 - freie Zeiteinteilung in Abstimmung mit dem Betreuer
 - Arbeiten finden am ITIV und FZI statt
- **Ziele**
 - Interdisziplinäre Projektarbeiten
 - Recherche / Bewertung / Einordnung / Umsetzung / Dokumentation
- **Ergebnispräsentation**
 - Schriftliche Ausarbeitung in Form eines Praktikumsberichts
 - Präsentation 3-5 Folien, Pitch



Automatischer Austausch von Gewichten in künstlichen neuronalen Netzwerken

Neuronale Netzwerke finden in verschiedensten Gebieten Anwendung. Von der Klassifizierung von Bildern unterschiedlicher Domänen bis hin zur Erkennung von Schnarchgeräuschen lässt sich die Verwendung in verschiedene Anwendungsbereiche gliedern. Eine Verwendung desselben Modells in verschiedenen Domänen ist ein Forschungsschwerpunkt, der vor Allem in Ressourcen-beschränkten Umgebungen Anwendung findet. Eine Reihe erster Gewichte und zwei divergierende Schichten, welche auf Basis der Aktivierungen der ersten Schichten getroffen wird könnten verschiedene Aufgaben mit der gleichen Laufzeit berechnen. In dieser Arbeit soll ein Skript erstellt werden, welches das automatisierte Training eines solchen doppelendigen Modells ermöglicht.

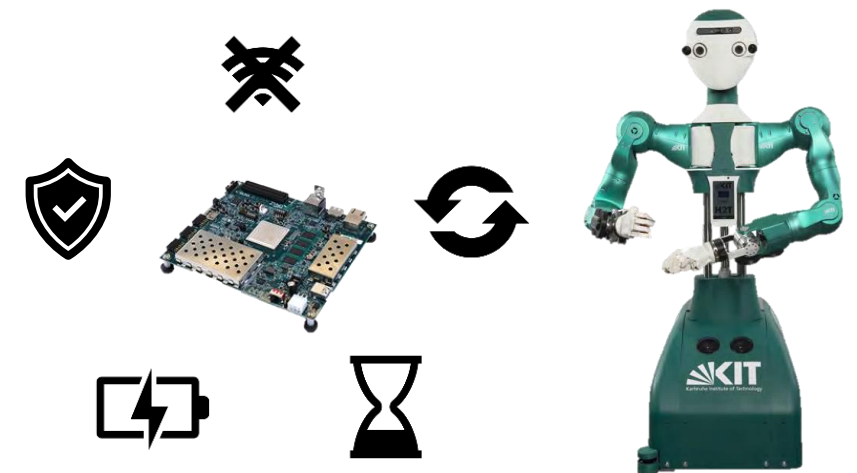
- Implementierung eines Tools zum Trainieren mehrerer Modelle mit gleicher Architektur auf unterschiedlichen Datensätzen
- Literaturrecherche über existierende Methoden zur parallelen Nutzung von ML-Modellen für verschiedene Tasks
- Voraussetzungen:
 - Gute Kenntnisse in Python und in ML-Bibliotheken
 - Gute Kenntnisse in Architektur und Funktionsweise verschiedener ML-Modelle



Assistive Robotics: Embedded Vision for Humanoid Robot

Artificial intelligence has entered a broad range of applications to make systems for, e.g., image classification or speech recognition, more precise and reliable than previous hand-optimized algorithms. One common method is supervised Deep Learning. In this process, deep neural networks learn to solve predefined tasks based on a large database with labelled examples. But, for artificial intelligence to enrich mobile applications, it must be embedded and adapted to the limited hardware constraints. In this internship, you will implement an embedded perception function for a humanoid robot, which is performed on RGB-data.

- Implementation of embedded vision into the humanoid robot
 - Finetuning of neural network for visual perception
 - Development of hardware accelerator
 - Integration of accelerator into robot framework
- Prerequisites
 - Basic knowledge in Python, C++
 - Basic understanding of neural networks

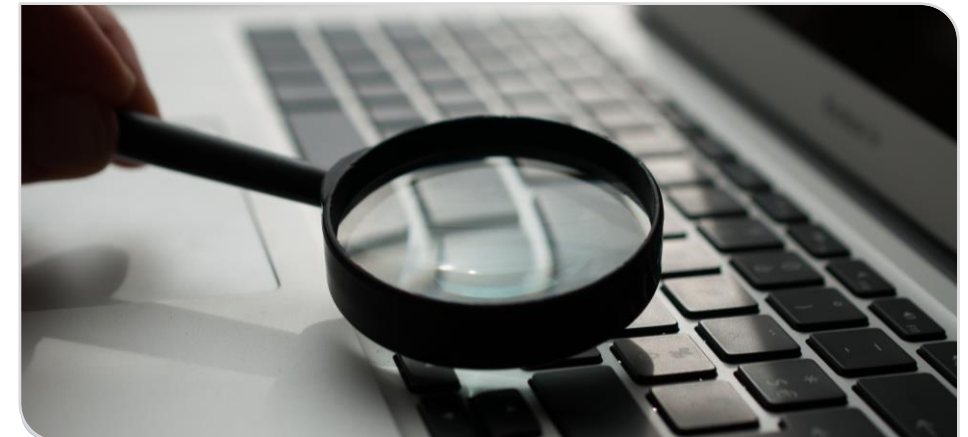


Topic available in German or English

Optimizing datagram encryption in embedded monitoring system

The increasing networking and interoperability of components in vehicles with the capability for autonomous driving lead to a constant exchange of information, for which source and destination cannot always be clearly determined due to the use of multiple bus systems. By using an extensible and abstract hardware library of different reconfigurable hardware components, which allows non-invasive monitoring of different communication systems, many different communication protocols can be monitored. Save the monitored datagrams encrypted is your task.

- Task: Integrating an encryption module into an embedded monitoring platform.
- Prerequisites:
 - Very good knowledge of VHDL, Verilog (through HSC, DHL, or similar lectures) or alternatively in HLS languages such as SystemC, SystemVerilog or Chisel
 - Good knowledge in C++



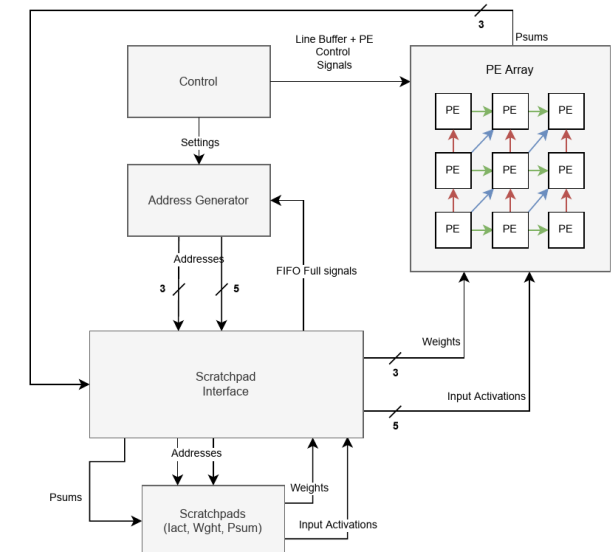
Entwicklung und Integration von KI-Beschleunigern für autonomes Fahren

- KI-Algorithmen für zukünftiges autonomes Fahren
- Besondere Anforderungen an Hardware
 - Energieeffizienz
 - Adaptivität / Update-Support
 - Mixed-Criticality-Systeme



Themen: Weiterentwicklung eines Beschleuniger-Designs

- Erweiterung für neue Netz- und Layertypen
- Integration in ML-Frameworks wie PyTorch & Tensorflow
- Prototypen-Entwicklung auf FPGA-Basis



Universal Prototyping Platform for Embedded Systems Research

We are conducting research on various topics around embedded systems. To demonstrate these topics we started to build a universal demonstrator platform. Based on the iRobot Create 3, it will consist of a wide variety of base blocks, that can be combined to construct a suitable setup. We are therefore looking for you to advance this platform and integrate our and your own ideas. The list below shows an example of the software, hardware and use cases we want to integrate in the future.

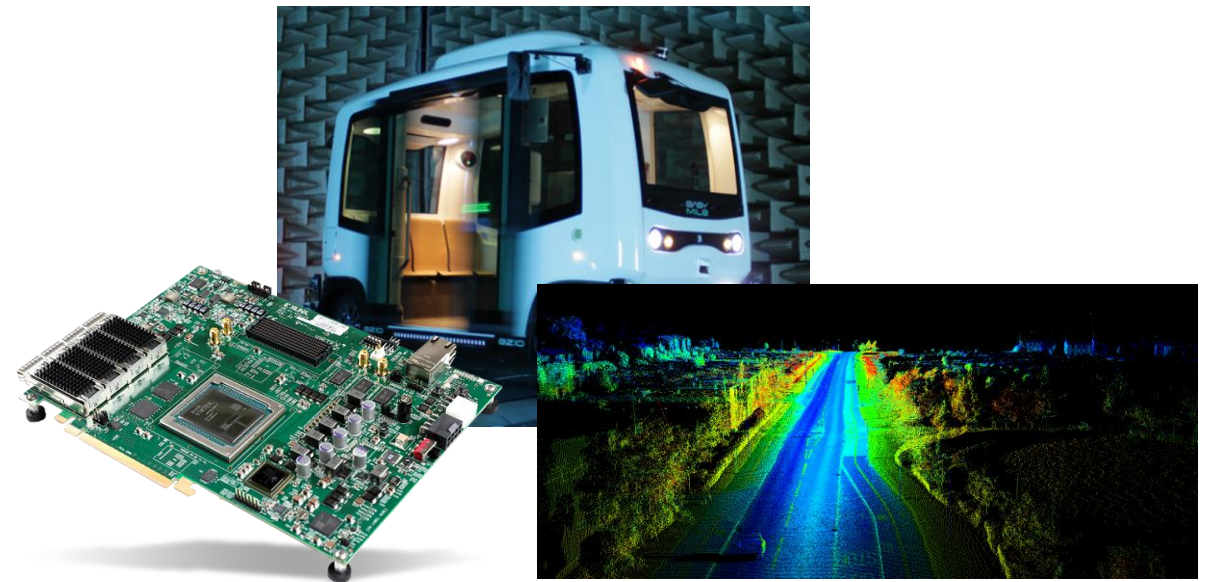
- Languages and frameworks
 - C++ / Rust / Python / ROS2
 - Verilog / VHDL
- Hardware and Sensors
 - Raspberry Pi, FPGA, Arduino
 - RGB/Event Camera, Lidar, ...
- Use Cases
 - Object Detection, SLAM, ...
 - Real-time Autonomous Functions, ...
 - Load Balancing & Migration



FPGA-based Sensor Data Processing

Autonomous vehicles use various sensors, such as Lidar, Radar or Cameras for environment perception. These sensors produce large amounts of data, which need to be processed fast and energy efficiently. Compute-intensive steps of this processing chain are accelerated with dedicated hardware accelerators such as FPGAs. We are developing FPGA-designs, for example for pointcloud sparsification and analytical or AI-based perception. Therefore we are looking for you to help us with design, testing and implementation.

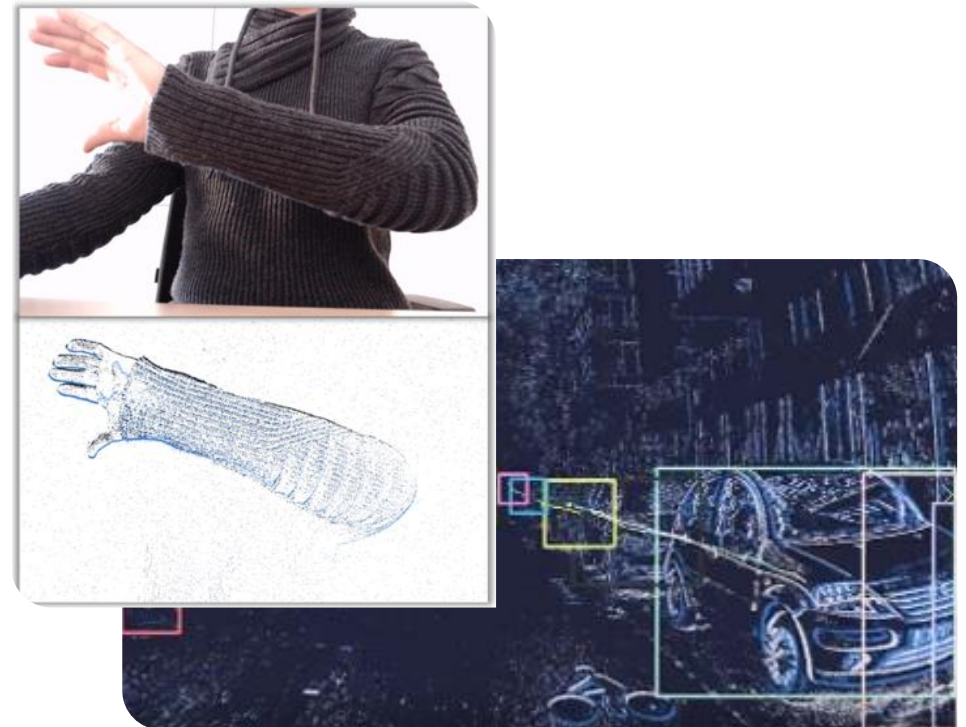
- Hardware Design
 - Verilog, VHDL or Chisel
 - Testing: cocotb or UVM
- Sensor data processing
 - AI-based and analytical methods
 - Lidar, Radar, Cameras
- Accelerators & Modules
 - Vector-Processors
 - AI-Accelerators: Gemmini, NVDLA, TVM-VTA
 - hls4ml



Neuromorphic Vision

Neuromorphic Engineering presents a paradigm shift in computing that takes close inspiration from biological brains. Akin to the impulse-based information processing in brains, neuromorphic systems compute on an abstraction of such impulses, referred to as events. This includes neuromorphic sensors such as event-based cameras which mimic the human eye. We develop such neuromorphic systems to perform action recognition and perception tasks.

- Software
 - Data preprocessing and networking
 - User interface development
- Hardware
 - Platform evaluation
 - VHDL design
- Machine Learning
 - Training of Spiking Neural Networks



Developing AI-powered software and hardware for medical decision making

Using sensors to monitor patient health - in real-time - is one of the most exciting areas of medical research. We need sophisticated AI algorithms to process and interpret enormous amounts of data accurately. In this project, we develop machine learning models to predict health outcomes and detect anomalies based on sensor data from wearables, medical devices, and other sources. The research topic includes analysis and interpretation of ECG, EEG, activity (accelerometer and gyroscope), respiratory, and sleep-related biosensor data.

As part of the project, you would also have the opportunity to work in your own area of interest - including ex. image processing or machine vision.

- AI and Neural Nets (Data processing pipelines, Model training, Evaluation)
- Vital signal analysis with State-of-the-Art wearable technologies
- Software development (Python AI Libraries, Docker, Cloud, Back-/Frontend)
- Hardware design and development (Req. Engineering, System design)



Aufbau eines Rechenclusters zur parallelen Datenverarbeitung

Daten, insbesondere Zeitreihen, werden in vielen Bereichen des alltäglichen Lebens immer wichtiger. Die Analyse dieser Daten erlaubt es Prognosen über den künftigen Verlauf zu treffen. So können historische Zeitreihendaten genutzt werden, um bspw. Aussagen über das künftige Kaufverhalten von Kund*innen, oder die Ausfallrate von Elektronikgeräten zu treffen. Ein wichtiger Aspekt der Zeitreihenanalyse ist die Mustererkennung. Mit dieser können sowohl typische als auch abweichende Verhaltensmuster erkannt werden. Hierfür soll ein Rechencluster zur parallelen Datenverarbeitung aufgebaut werden, der als Demonstrator dienen soll.

■ Aufgaben

- Verarbeitung und Analyse großer Datenmengen
- Aufbau eines Demonstrator-Rechenclusters (Hardware und Software)
- Visualisierung eines Use Cases für Demonstrationen

■ Voraussetzung

- Unix Kenntnisse von Vorteil
- Du arbeitest selbständig und strukturiert, bist motiviert und engagiert.

